5

CLAIMS

What is claimed is:

- 1. A method of maintaining coherency in a multi-processor-bus computer system comprising the acts of:
 - (a) receiving a request at a host controller from a first bus, the request having a corresponding first address;
 - (b) initiating a first read to a first portion of memory corresponding to the first address;
 - (c) snooping a second bus for the first address;
 - (d) determining the next snoop transaction to be processed, the next snoop transaction having a corresponding second address; and
 - (e) searching a posting queue for a posted writeback, the posted writeback having the corresponding second address, wherein the act of searching occurs simultaneously with respect to the act of snooping.
 - 2. The method of maintaining coherency, as set forth in claim 1, further comprising the acts of:

- (a) detecting a posted writeback having the corresponding second address in the posting queue;
- (b) re-ordering the posting queue such that the posted writeback to the second address is moved up in the posting queue; and
- (c) initiating a second read to a second portion of memory corresponding to the second address.
- 3. The method of maintaining coherency, as set forth in claim 2, wherein the acts of detecting, re-ordering and initiating a second read occur simultaneously with respect to the act of snooping.
- 4. The method of maintaining coherency, as set forth in claim 1, wherein act (a) comprises the act of receiving a request at a host controller from a processor bus.
- 20 5. The method of maintaining coherency, as set forth in claim 1, wherein act (a) comprises the act of receiving a request at a host controller from an input/output (I/O) bus.

- 6. The method of maintaining coherency, as set forth in claim 1, wherein the act of snooping is performed by a coherency control module.
- 5 7. The method of maintaining coherency, as set forth in claim 1, wherein act (d) comprises the acts of:
 - (a) searching a snoop queue in a processor controller corresponding to the second bus; and
 - (b) returning a next snoop forecast signal from the processor controller to the coherency control module, the next snoop forecast signal comprising the corresponding second address.
 - 8. A method of predicting a future snoop transaction comprising the acts of:
 - (a) initiating a snoop request having a corresponding first address from a coherency control module to a processor controller;
 - (b) returning a snoop result corresponding to the first address from the processor controller to the coherency control module; and

- (c) returning a next snoop forecast signal having a second address and corresponding to
 the next snoop request to processed from the processor controller to the coherency
 control module.
- 9. The method of predicting a future snoop transaction, as set forth in claim 8, further comprising the acts of:
 - (a) detecting a posted writeback having the corresponding second address in the posting queue;
 - (b) re-ordering the posting queue such that the posted writeback to the second address is moved up in the posting queue; and
 - (c) initiating a second read to a second portion of memory corresponding to the second address.
- 10. The method of predicting a future snoop transaction, as set forth in claim 9,
 wherein the acts of detecting, re-ordering and initiating a second read occur simultaneously with
 respect to the act of snooping.

- 11. The method of predicting a future snoop transaction, as set forth in claim 8, wherein acts (b) and (c) occur simultaneously.
- 12. A method of maintaining cache coherency in a multi-processor-bus computer system comprising the acts of:
 - (a) reading a snoop transaction queue comprising at least a first snoop request having a first address and a second snoop request having a second address, the second snoop request being prioritized subsequent to the first snoop request in the snoop transaction queue; and
 - (b) processing the first snoop request while simultaneously searching a posting queue for the second address corresponding to the second snoop request.
- 13. The method of maintaining a cache coherency in a multi-processor-bus computer system, as set forth in claim 12, further comprising the acts of:
- 20 (a) detecting a posted writeback having the corresponding second address in the posting queue;

- (b) re-ordering the posting queue such that the posted writeback to the second address is moved up in the posting queue; and
- (c) initiating a second read to a second portion of memory corresponding to the second address.
- 14. The method of maintaining cache coherency in a multi-processor-bus computer system, as set forth in claim 13, wherein the acts of detecting, re-ordering and initiating a second read occur simultaneously with respect to the act of snooping.
- 15. The method of maintaining cache coherency in a multi-processor-bus computer system, as set forth in claim 12, wherein act (a) comprises the acts of:
 - (a) searching the snoop queue in a processor controller; and
 - (b) returning a next snoop forecast signal from the processor controller to a coherency control module, the next snoop forecast signal having a second address and corresponding to the second snoop request.

- 16. A coherency control module configured to snoop a bus for a first address while simultaneously searching a posting queue for a second address.
- 5 17. The coherency control module, as set forth in claim 16, comprising a first request module configured to receive requests from a first processor controller.
 - 18. The coherency control module, as set forth in claim 17, comprising a second request module configured to issue snoops to a second processor controller.
 - 19. The coherency control module, as set forth in claim 18, wherein the posting queue resides in the second processor controller.
 - 20. The coherency control module, as set forth in claim 16, wherein the coherency control module is configured to snoop a processor bus for a first address.
 - 21. The coherency control module, as set forth in claim 16, wherein the coherency control module is configured to snoop an input/output (I/O) bus for a first address.